

# Machine Learning in the calibration process of MercuryDPM Discrete Particle Model

Quang Hung Nguyen

Head Supervisor: Thomas Weinhart, Daily Supervisor: Anthony Thornton, Additional member: Chen Kuan.  
Multi-Scale Mechanics Group, Faculty of Engineering Technology, University of Twente.

## 1 Introduction



Figure 1: Examples of Granular Materials.[1]

Granular material is a family of material characterized by its large bulk of densely packed particles, ranging from nanometers to centimeters [2], and is able to resist deformation and form heaps, i.e., behave like a solid and withstand strong shear force [3]. Simple examples of granular materials include sand, gravel, clays, seeds, nuts, and all ranges of powders such as coffee powder, cement powder, which is shown in figure 1. Furthermore, many processes and equipments in chemical plants use granular materials, such as catalysis, adsorption, and heat exchangers. Granular materials are projected to make about half of the products and three-quarters of the raw materials used in the chemical industry [4]. Thus, understanding how granular materials behave is of great significance.

Granular material's bulk mechanical behavior is simulated using a Discrete Particle Model (DPM, or Discrete Element Method - DEM), which generates the movement of individual particles to capture the macro-scale behavior. The DPM is a family of numerical methods for computing the motion of a large number of particles [5]. Since the properties of granular materials differ wildly, these simulations require an extensive calibration process designed individually for each type of granular material. Some parameters of the granular material model can be measured directly, such as size distribution or density. However, other parameters are effective parameters (i.e., they result from a simplified particle model) and thus cannot be directly measured. These parameters are then calibrated by choosing a few standard calibration setups (rotating drum, heap test, ring shear cell) and simulating these setups in a DPM simulation, and the missing parameters are determined such that the response of the experimental and simulation setups match. This has been done using a Convolutional Neural Network

[6] and Genetic Algorithm [7]. However, these methods are very costly in terms of time and resources, since it requires a large number of DEM simulations. In this research assignment, another algorithm will be studied: GrainLearning - a Python-based Bayesian calibration tool developed by H.Cheng et al. [8]. GrainLearning utilizes the recursive Bayesian algorithm to estimate the uncertainty parameters in DPM. As mentioned above, each model requires extensive calibration, and using GrainLearning can make the calibration process more effective by iteratively searching for the optimized parameters, effectively reducing the number of DEM simulations required.

GrainLearning has been implemented in MercuryDPM and produces satisfactory results in many cases. However, it is far from clear whether the chosen optimization algorithm is optimal, i.e., whether the calibration results in an optimal parameter set and whether the optimum could be reached faster since currently, a calibration process could run for hours. Therefore, the research question here is that: Is there a more efficient way to calibrate the parameters for DEM simulation using GrainLearning?

## **2 Calibration**

### **2.1 DEM Simulations**

MercuryDPM is an open-source DEM software package, developed by Thomas et al.[9]. The packag

### **2.2 GrainLearning**

### **2.3 Artificial Neural Network**

Artificial Neural Network (ANN) is a set of algorithms that seeks to identify correlations in data utilizing a technique that inspired by the way human brain operates - mimicking how each neurons in the brain signaling each other. The most basic ANN models is the Feedforward Multilayer Perceptron Neural Network (MLPNN), in which the purpose is to define the mapping between the input and output  $y = f(x; \theta)$ , and approximate the parameter  $\theta$  which results in the best possible function. In MLPNN, the data will flows in one direction from the input to output, hence the name feedforward. Like other supervised learning algorithms, a MLPNN need to be trained before it can accurately describe the relations between the input and output. This is typically done by feeding the network with pre-labeled data, compare the model's output with the desired output, and update the weights parameter  $\theta$  - a process called backpropagation. In the context of this Assignment, ANN will be used when refering to Feedforward Multilayer Perceptron Neural Network.

## References

- [1] A. Fernandes, H. Gomes, E. M. B. Campello, P. Pimenta, A fluid-particle interaction method for the simulation of particle-laden fluid problems, 2017. doi:10.20906/CPS/CILAMCE2017-0139.
- [2] A. B. Yu, F. Bassani, G. L. Liedl, P. Wyder, Powder Processing: Models and Simulations, Encyclopedia of Condensed Matter Physics, Elsevier, Oxford, 2005, pp. 401–414. doi:10.1016/B0-12-369401-9/00556-8.
- [3] E. S. Oran, J. P. Boris, R. A. Meyers, Encyclopedia of Physical Science and Technology (Third Edition), Academic Press, New York, 2002, Ch. Fluid Dynamics, pp. 31–43. doi:10.1016/B0-12-227410-5/00248-9.
- [4] R. M. Nedderman, Statics and Kinematics of Granular Materials, Cambridge University Press, Cambridge, 1992, Ch. Introduction, pp. 1–6. doi:10.1017/CB09780511600043.002.
- [5] X. Weng, Modeling of complex hydraulic fractures in naturally fractured formation, Journal of Unconventional Oil and Gas Resources 9 (2015) 114–135. doi:10.1016/j.juogr.2014.07.001.
- [6] L. Benvenuti, C. Kloss, S. Pirker, Identification of dem simulation parameters by artificial neural networks and bulk experiments, Powder Technology 291 (2016) 456–465. doi:10.1016/j.powtec.2016.01.003.
- [7] H. Q. Do, A. M. Aragón, D. L. Schott, A calibration framework for discrete element model parameters using genetic algorithms, Advanced Powder Technology 29 (6) (2018) 1393–1403. doi:10.1016/j.appt.2018.03.001.
- [8] H. Cheng, T. Shuku, K. Thoeni, H. Yamamoto, Probabilistic calibration of discrete element simulations using the sequential quasi-monte carlo filter, Granular Matter 20 (1) (2018) 11. doi:10.1007/s10035-017-0781-y.
- [9] T. Weinhart, L. Orefice, M. Post, M. P. van Schrojenstein Lantman, I. F. Denissen, D. R. Tunuguntla, J. Tsang, H. Cheng, M. Y. Shaheen, H. Shi, P. Rapino, E. Grannonio, N. Losacco, J. Barbosa, L. Jing, J. E. Alvarez Naranjo, S. Roy, W. K. den Otter, A. R. Thornton, Fast, flexible particle simulations — an introduction to mercurydpm, Computer Physics Communications 249 (2020) 107129. doi:10.1016/j.cpc.2019.107129.